



# OPEN Study of conveyor belt deviation detection based on improved YOLOv8 algorithm

Yunfeng Ni, Haixin Cheng✉, Ying Hou & Ping Guo

Conveyor belt deviation is a common and severe type of fault in belt conveyor systems, often resulting in significant economic losses and potential environment pollution. Traditional detection methods have obvious limitations in fault localization precision and analysis accuracy, unable to meet the demands of efficient and real-time fault detection in complex industrial scenarios. To address these issues, this paper proposes an improved detection algorithm based on YOLOv8, aiming to achieve efficient and accurate detection during the operation of the belt. Firstly, an Enhanced Squeeze-and-Excitation (ESE) module is incorporated into C2f to boost feature extraction for rollers and belts. Secondly, the construction of the BiFPN\_DoubleAttention module in the neck network enhances bidirectional feature fusion and attention mechanism, thereby improving multi-scale object localization accuracy under complex environments. Then, a Multi-Head Self-Attention (MHSA) mechanism is introduced in the head network, better capturing positional features of small roller targets and belt areas in various environments, thus enhancing detection performance. Finally, extensive experiments are conducted on a self-built dataset, achieving an accuracy of 98.1%, mAP0.5 of 99.0%, and a detection speed of 46 frames per second (FPS). This method effectively handles variations and disturbances in the conveyor belt transportation environment, meeting real-time diagnostic needs for belt misalignment in the industry.

**Keywords** Conveyor belt deviation, YOLOv8, EffectiveSE, BiFPN\_DoubleAttention, MHSA, Deviation judgment method

The conveyor belt is an essential piece of equipment in modern secure production and transport, providing continuous transit of bulk materials. It primarily consists of belts, rollers, racks, tensioning devices, transmission devices, and other auxiliary devices. The conveyor belt offers several advantages, including large conveying volume, long transport distance, stable operation, low power consumption, and easy loading and unloading. It is widely used in various industries such as coal mining, construction sites, power supply, and metallurgy. With the expansion of industrial production scale, the carrying capacity of the conveyor belt has increased, bringing substantial economic benefits to the material handling industry. As the conveyor belt accounts for 40–60% of the total cost of the conveyor, continuous improvement in its design and performance is crucial. The maximum length of the conveyor can reach several kilometers, and its running status directly affects the stability of the entire production and transportation process<sup>1</sup>.

The harsh and complicated working environment of conveyor belts often leads to deviation faults<sup>2</sup>. Conveyor belt deviation refers to the belt running process where the conveyor belt surface center line deviates from the frame center line, gradually running to one side. According to statistics, conveyor belt stoppage, production stoppage, and other accidents caused by belt deviation account for about 10–30%. The fault of conveyor belt deviation results in belt wear, material spillage, wasted resources, and environmental pollution. Severe cases can cause fires, injuries, and property damage<sup>3</sup>. Moreover, if the conveyor belt deviation problem is not well solved, it restricts the wider application of belt conveyors, impeding their development.

With the advancement of state detection technology, conveyor belt runout detection methods have evolved from contact methods to non-contact methods. The contact type is mainly divided into three: manual inspection, sensor-based, and mechanical measurement<sup>4</sup>. The manual inspection method is labor-intensive, and detection accuracy decreases over time. Sensor-based methods typically require deviation sensors<sup>5</sup>, symmetrically mounted on the frame on either side of the conveyor belt to activate travel switches. When the conveyor belt tilts, the edge pushes the deviation sensors to trigger the travel switches to send out deviation correction signals, which are fed back to the terminals to complete conveyor belt deviation correction detection. Mechanical measurement

Affiliation College of Xi'an University of Science and Technology, Xian, China. ✉email: cxj2025lucky@163.com

methods include installing correction rollers or emergency switches for the conveyor belt, but they only work when the conveyor belt touches the vertical rollers or emergency switches. The structure and principle of this method are simple, but it causes wear on the conveyor belt edges and poor detection accuracy.

Non-contact detection methods, which incorporate machine vision<sup>6–8</sup> algorithms into conveyor belt fault prediction, obtain relevant fault information from the conveyor's operating state and predict future fault trends<sup>9</sup>. Many high-performing methods utilize the Canny edge detection algorithm<sup>10</sup> for traditional machine vision techniques, extracting belt edge features. Liu<sup>11</sup> proposed an approach using edge detection and line fitting to extract the belt edge, proposing an analysis method to quickly determine if the belt has deviated from its path. Wang<sup>12</sup> divided the conveyor belt into regions of interest, extracted straight line information of the conveyor belt edges through image enhancement, wavelet transformation, Canny edge detection, and Hough transform, and conducted belt deviation detection with a reference line of a normal running conveyor belt. In recent years, In recent years, deep learning has been broadly applied to machine vision for image recognition, target detection, and classification<sup>13</sup>. More scholars are using deep learning's target detection to solve conveyor belt runout detection in complex environments<sup>14,15</sup>.

Wlodarczyk-Sielicka<sup>16</sup> introduced a novel method for tape edge detection utilizing deep convolutional networks, FCN (Fully Convolutional Networks)<sup>17</sup>, DeepLab<sup>18</sup>, HED (Holistically-Nested Edge Detection)<sup>19</sup>. This method offers reliable target detection and anti-jamming capabilities, addressing the shortcomings of untimely processing in traditional mechanical anti-bias methods and the diminished accuracy of machine vision for strip edge detection. However, this approach incurs a relatively large deviation degree (DD) error. Unlu<sup>20</sup> employed a real-time belt detection algorithm based on multi-scale feature fusion networks, which improves accuracy and real-time performance through the use of deep divisible convolution to lightweight the model and feature pyramid networks (FPNs) for information fusion from different feature layers. Still, given image segmentation's pixel-level operation, performance tops out at 13.4 FPS. Zhang<sup>21</sup> proposed a new deep learning-based method for tape deviation monitoring, integrating CSPDarknet and Spatial Pyramid Pooling (SPP) as the backbone extraction network of YOLOv5<sup>22–24</sup>, boosting straight line detection and improving belt edge detection accuracy. This addresses complex background strip edge detection challenges. Sun<sup>25</sup> developed a system for curved belt deviation assessment using an ARIMA-LSTM combined prediction model and the OC-SVM algorithm, enabling real-time detection, prediction, correction, and warning of strip abnormal deviation. Wang<sup>26</sup> introduced a variable step size row anchor segmentation method employing the UFLD algorithm, incorporated a Convolutional Block Attention Module (CBAM) for enhanced feature extraction, and improved the convolutional and downsampling operations in the ResNet-18 Stem and residual modules to better detect conveyor belt edges.

In summary, this paper aims to accurately detect conveyor belt runout issues in complex environments and ensure safe material transport. It utilizes an improved YOLOv8<sup>27–29</sup> algorithm combined with a conveyor belt runout discrimination criterion to determine transport state safety. Firstly, the paper adopts the EffectiveSE mechanism based on the YOLOv8 target detection network structure, enhancing attention to target region feature information and improving detection accuracy for fuzzy and small target regions. Second, the BiFPN\_DoubleAttention module is introduced for bidirectional feature fusion and weighted rectification on multiple feature scales, enabling better combination of low-level detail and high-level semantic information to improve detection accuracy and robustness. Finally, the P5/32-Large is equipped with the MHSA Multi-Head Self-Attention Mechanism, allowing better understanding of global structure and relative positional relationships for small (rollers) and larger (conveyor belt area) target objects, improving detection performance. Compared to the traditional YOLOv8 algorithm, this method performs better for roller and conveyor belt detection, enhancing model accuracy, robustness, and application performance.

## Materials and methods

### YOLOv8 Model Architecture

YOLOv8 is a cutting-edge SOTA model, built upon previous versions of the YOLO series, introducing significant improvements and innovations to enhance performance and flexibility, making it the best choice for tasks such as image classification, object detection, and instance segmentation. The model proposes a new backbone network, a new Anchor-Free Detection header, and a new loss function, operating efficiently on a wide range of hardware platforms. The network structure of YOLOv8 consists mainly of a backbone, a neck, and a head.

Backbone also adopts the concept of the CSP module and replaces the C3 module in Yolov5 with the C2f module, which combines C3 and ELAN<sup>30</sup> in YOLOv7<sup>31,32</sup>, adopting the gradient diversion connection, enriching the information flow of the feature extraction network while maintaining its lightness. The SPPF (Spatial Pyramid Pooling Fast) module in Yolov5 is utilized, connecting each layer in series by sequentially connecting three sizes and 3 max pools, improving the accuracy of object detection at different scales and reducing the computational workload and latency.

The neck utilizes the PAN-FPN<sup>33</sup> structure to enhance the fusion of features of differing dimensions and generate a feature pyramid. This structure comprises two parts: the feature pyramid network (FPN) and the path aggregation network (PAN). The FPN enables the bottom layer of the feature map to contain stronger image semantic information via top-down upsampling, while the PAN enables the top layer to contain image position information via bottom-up downsampling. The two parts of the features are finally fused, combining the FPN and PAN structures to realize the complementarity of shallow position information and deep semantic information, achieving feature diversity and completeness, and improving recognition performance.

The head is the predicted output part of the whole network. Unlike the coupled head of YOLOv5, YOLOv8 adopts a decoupled head design. The decoupled head uses two independent branches for target classification and prediction bounding box regression, with different loss functions for each task. Binary cross-entropy loss (BCE loss) is used for classification, while distribution focal loss (DFL)<sup>34</sup> and CioU<sup>35</sup> are used for bounding

box regression. This detection structure improves accuracy and accelerates model convergence. Additionally, YOLOv8 is an anchorless detection model that directly predicts the object center, rather than predicting the distance from a known anchor frame. This reduces the number of frame predictions and accelerates the non-maximum suppression (NMS) process.

Different detection scene requirements divide YOLOv8 into five versions: YOLOv8n<sup>36–38</sup>, YOLOv8s<sup>39</sup>, YOLOv8m<sup>40</sup>, YOLOv8l<sup>41</sup>, and YOLOv8x<sup>42</sup>. Each version corresponds to different network depths and widths. Considering the conveyor belt runout detection model's high precision and lightweight, this paper selects the YOLOv8n mesh, which has a small volume and high precision.

### Improved methodology

Traditional conveyor belt deflection detection methods were inaccurate, so this paper introduces a YOLOv8-based deflection detection model with improved backbone, neck, and head. The network structure of the improved algorithm is shown in Fig. 1.

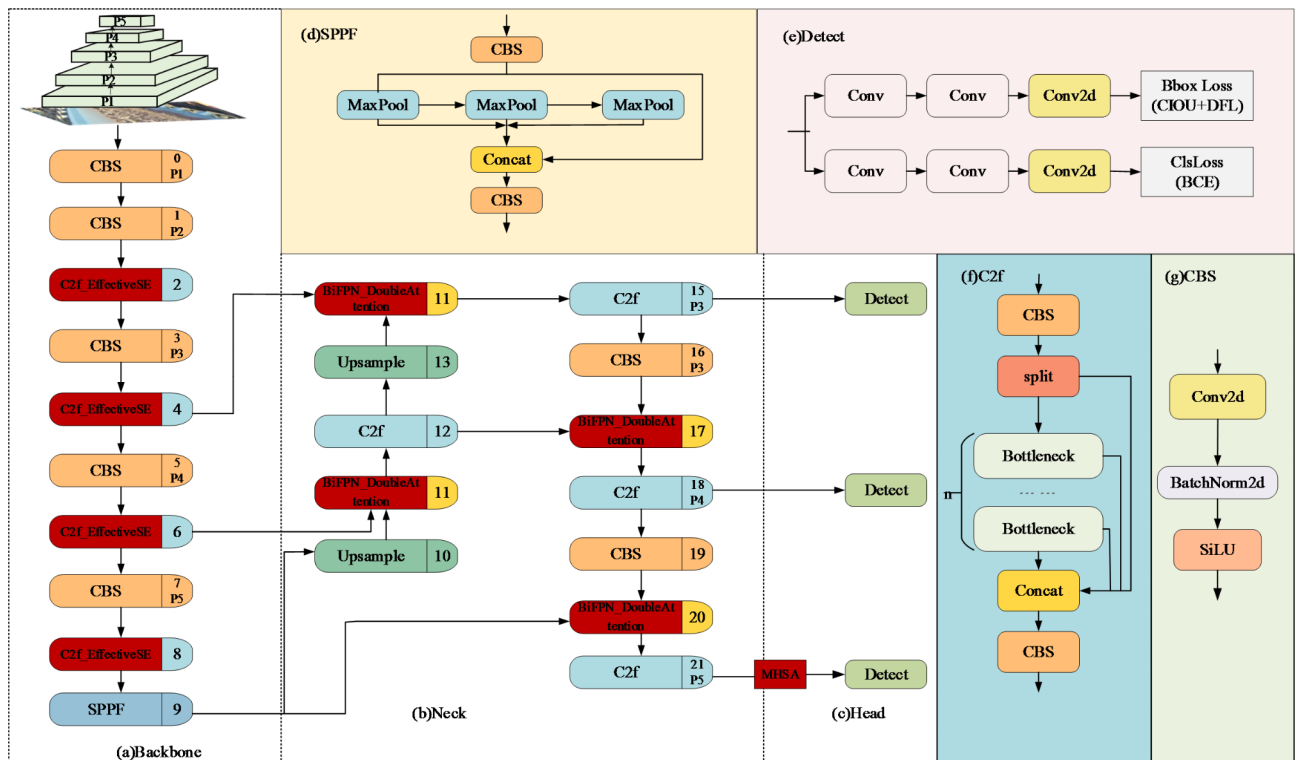
#### Backbone

The C2f module is a vital component of the YOLOv8 backbone network, which enhances feature representation and multi-scale information fusion through branching and connecting operations of the feature map. In this paper, we incorporate the EffectiveSE (Effective Squeeze and Extraction, ESE)<sup>43</sup> module into the C2f module, further improving the model's feature selection and weight adjustment capabilities. This results in accurate detection of rollers and belts in complex, dynamically changing real-world environments such as mines, sandy fields, and quarries.

EffectiveSE is a convolutional neural network structure for image classification, an improved version of SENet (Squeeze-and-Excitation Networks)<sup>44</sup>. The Squeeze-Excitation (SE) channel attention module in SENet was found to inadvertently reduce computational efficiency due to the use of two fully connected (FC) layers in the SE module design. To address this, the SE module is redesigned as ESE by replacing these two FC layers with one that maintains the channel dimension, thus avoiding loss of channel information and improving performance. The ESE process is defined as:

$$A_{eSE}(X_{div}) = \sigma(W_C(F_{gap}(X_{div}))) \tag{1}$$

$$X_{refine} = A_{eSE}(X_{div}) \otimes X_{div} \tag{2}$$



**Fig. 1.** The architecture of improved YOLOv8 network structure. The red boxes represent improvement points. C2f is used for concatenating different feature maps and other operations; CBS consists of Conv, BatchNorm, and SiLu activation functions for feature extraction; Spatial Pyramid Pooling Fast (SPPF) is used to increase feature diversity.

where  $X_{div} \in R^{C \times W \times H}$  is the diversified feature map calculated by  $1 \times 1$  conv in the OSA module. As a channel attentive feature descriptor, the  $A_{cSE} \in R^{C \times 1 \times 1}$  is applied to the diversified feature map  $X_{div}$  to make the diversified feature more informative. Finally, when using the residual connection, the input feature map is added to the refined feature map  $X_{refine}$ . The details of how the eSE module is plugged into the OSA module, are displayed in Fig. 2.

*Neck*

In the neck network structure of YOLOv8, we introduced the BiFPN\_DoubleAtten module, which combines BiFPN\_Concat2 and DoubleAttention. BiFPN\_Concat2 enables weighted fusion of multi-layer features, effectively consolidating feature information from different scales. The DoubleAttention mechanism adjusts fused features with attention, enhancing the ability to focus on key regions. Overall, this module optimizes multi-scale feature fusion and significantly improves the capability to accurately detect multi-scale target positions in complex environments, including rollers and conveyor belts.

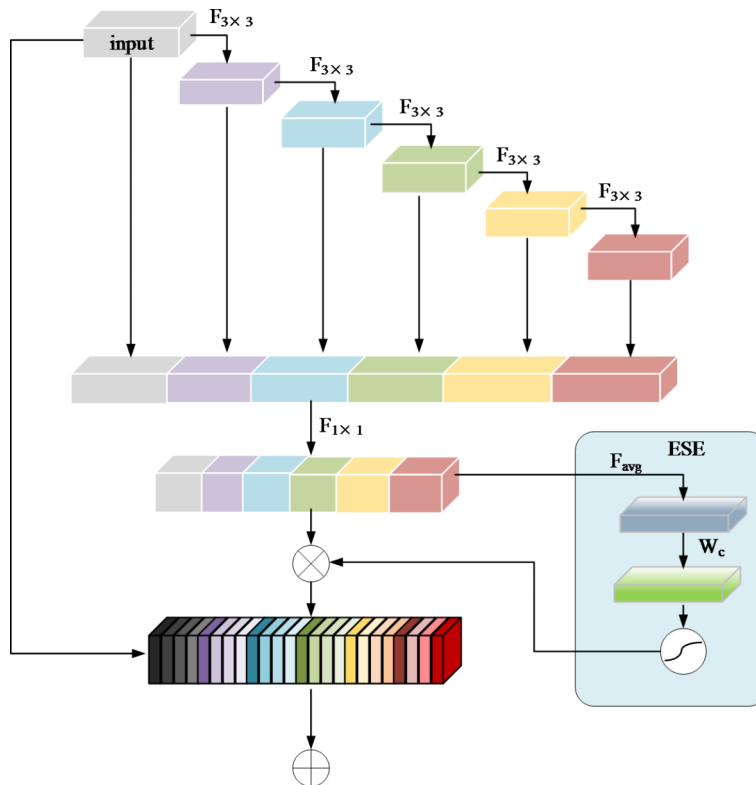
BiFPN<sup>45</sup> is a target detection optimization module that is built upon the PANet structure. It is designed to reduce complexity by discarding small contributions and using a single input edge. Additionally, it assigns increasing input feature weights to differentiate the importance of features in the fusion process, thereby enhancing the multi-scale target feature fusion effect, as shown in Fig. 3.

The BiFPN module adopts fast normalized fusion to avoid the unstable training caused by boundless fusion in weighted feature fusion, and the significant slowdown of GPU hardware caused by Softmax-based fusion. The calculation formula is as follows:

$$O = \sum_i \frac{w_i}{\sum_j w_j} \cdot I_i \tag{3}$$

where the ReLU activation function is added after each  $w_i$  to ensure that  $w_i \geq 0, \epsilon = 0.0001$  is used to avoid numerical instability, and the value of each normalized weight also falls between 0 and 1. Finally, BiFPN integrates bidirectional cross-scale connectivity and fast normalized fusion to obtain the final weighted feature pyramid network.

The attention mechanism is a process that imitates the human visual system, enabling a neural network to focus on the most relevant parts of the input data as it is processed. In computer vision, this means the network can



**Fig. 2.** The overall architecture of the ESE module.  $\otimes$  indicates element-wise multiplication and  $\oplus$  denotes element-wise addition.

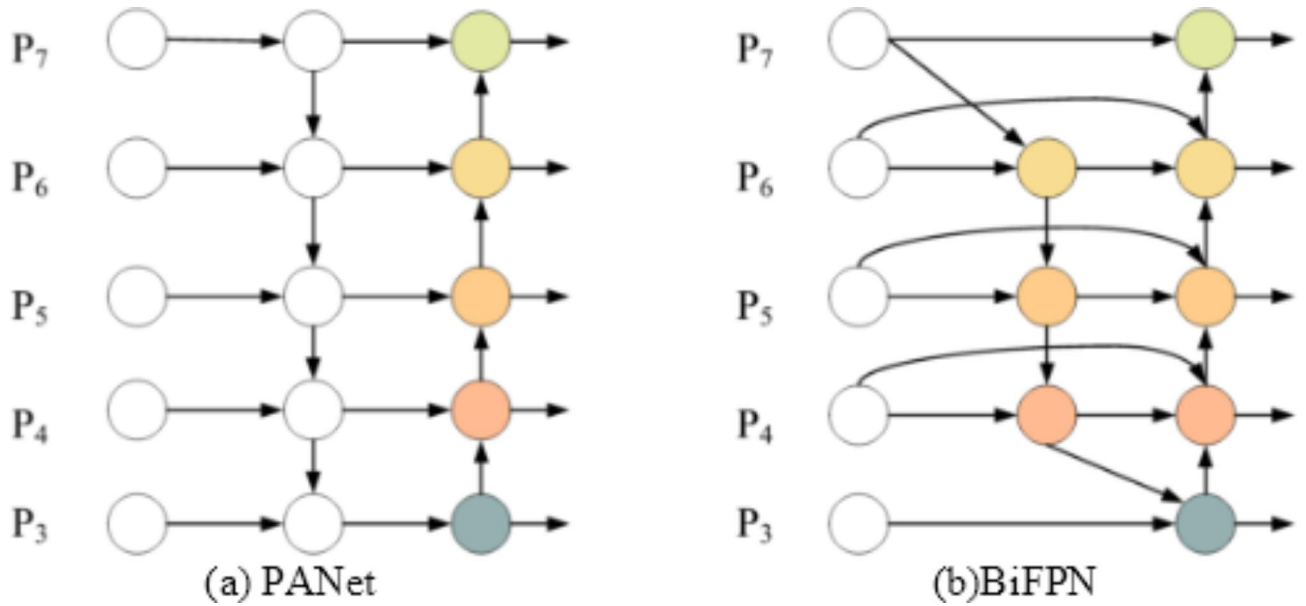


Fig. 3. The network architecture of PANet and BiFPN. (a) PANet adds an additional bottom-up pathway on top of FPN; (b) BiFPN structure is based on PAN and uses the fast normalized fusion method for fusion with weights with better accuracy and efficiency trade-offs.

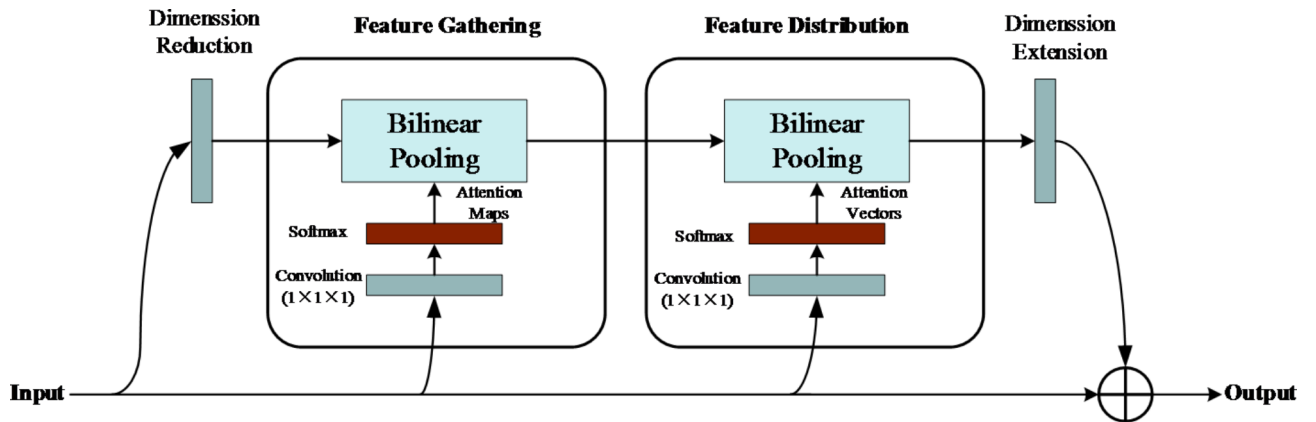


Fig. 4. The computational graph of the proposed double attention block.

dynamically select which parts of an image to focus on to improve performance on a task. Double Attention Networks (DAN)<sup>46</sup> is a neural network architecture for computer vision tasks designed to efficiently capture both global and local information in an image to improve task performance (as shown in Fig. 4). It achieves this through the introduction of two attention modules, one for global information focusing on the overall structure of the image and one for local information focusing on local details. This architecture makes full use of the different information in the image within a deep neural network and is formulated as follows:

$$\begin{aligned}
 Z &= F_{dister}(G_{gather}(X), V) \\
 &= G_{gather}(X) \text{softmax}(\rho(X; W_\rho)) \\
 &= [\phi(X; W_\phi) \text{softmax}(\theta(X; W_\theta))^T] \text{softmax}(\rho(X; W_\rho))
 \end{aligned}
 \tag{4}$$

**Head**

After the last C2f module in the Head part of the YOLOv8 network structure, the Multi-Head Self-Attention (MHSA)<sup>47</sup> Mechanism is added to the P5/32-large layer to enhance the model’s ability to express detailed features in complex scenes, enabling better capture of positional features for small target rollers and large target conveyor from complex and dim backgrounds.

The Multi-Head Self-Attention (MHSA) module is a stack of N single-head self-attention modules, with multi-head attention forming a central component in the Transformer encoder structure. The feature mapping

sequence is normalized and passed to the multi-head attention layer, which consists of single-head self-attention. Using a cubic linear mapping, each element is weighted according to the values of other elements. The single-head attention mechanism achieves sequence weighting by combining each element into query and key vectors, which undergo matrix multiplication, scaling, masking, and SoftMax operations, and are weighted by value vectors. Figure 5 shows the structure of the MHSA Multihead Self-Attention Mechanism.

In the multi-head attention mechanism, the input sequence is first converted into query vectors, key vectors, and value vectors. Each head can learn different query vectors, key vectors, and value vectors. Then, each head computes the similarity between each query vector and each value vector and computes the corresponding weights. These weights are used to perform a weighted sum on the value vectors to generate an output sequence for each head. Finally, these output sequences are spliced together to form the final output sequence. The multi-head attention mechanism can effectively improve model performance and enable better capture of relevant information in input sequences. The MHSA multiple-head attention output formula is:

$$MultiHead(Q, K, V) = Concat(head_1, \dots, head_h)W^O$$

$$head_i = Attention(QW_i^Q, KW_i^K, VW_i^V) \tag{5}$$

where  $W^O, W_i^Q, W_i^K$  and  $W_i^V$  is the weight matrix; Satisfy  $W^O \in R^{hd_k \times d_{mod\ el}}$ ,  $W_i^Q \in R^{d_{mod\ el} \times d_Q}$ ,  $W_i^K \in R^{d_{mod\ el} \times d_K}$ , and  $W_i^V \in R^{d_{mod\ el} \times d_v}$  respectively;  $h$  is the number of multiple attentions;  $d_{mod\ el}$  and  $d_v$  are the dimensions of model and  $V$ .

### Experiments

#### Experimental environment

All experimental environments are based on the Linux operating system, with an Intel(R) Xeon(R) CPU E5-2680 v4 CPU, an NVIDIA GeForce RTX 4090 GPU, and 32GB of RAM. The programming environment utilizes the deep learning framework PyTorch version 1.8.0, implemented in Python 3.8. During model training, the number of epochs is set to 300, the batch\_size is set to 32, and the initial learning rate is set to 0.01.

#### Datasets and preprocessing

To validate the effectiveness of the method, experiments were conducted on a self-constructed dataset. A total of 5,800 images were collected, and categorized into three types: underground mines, sand fields, and ore yards. The coal mine data consisted of 4,241 images, with 3,493 images self-collected from several coal mines and 748 images selected from public datasets. Additionally, 1,000 images of sand fields and 559 images of ore yards were selected from public datasets. These data included various working scenarios such as empty and loaded conditions, as well as day and night settings. The data were labeled with “0” for drum labeling and “1” for conveyor belt labeling. The dataset was then divided into training, testing, and validation sets in a 7:2:1 ratio. Figure 6 illustrates a portion of the dataset.

#### Evaluation index

To assess the detection performance of the model, four indicators were used to evaluate the experimental results: Precision (P), Recall (R), F1-Score, and Mean Average Precision (mAP).

In the target detection test, TP (true positive) denotes a positive sample predicted by the model to be in the positive category, FP (false positive) denotes a negative sample predicted by the model to be in the positive category, and FN (false negative) denotes a positive sample predicted by the model to be in the negative category.

Precision is the ratio of true positive predictions to the total number of samples detected, and is used to assess model accuracy. The equation for precision is:

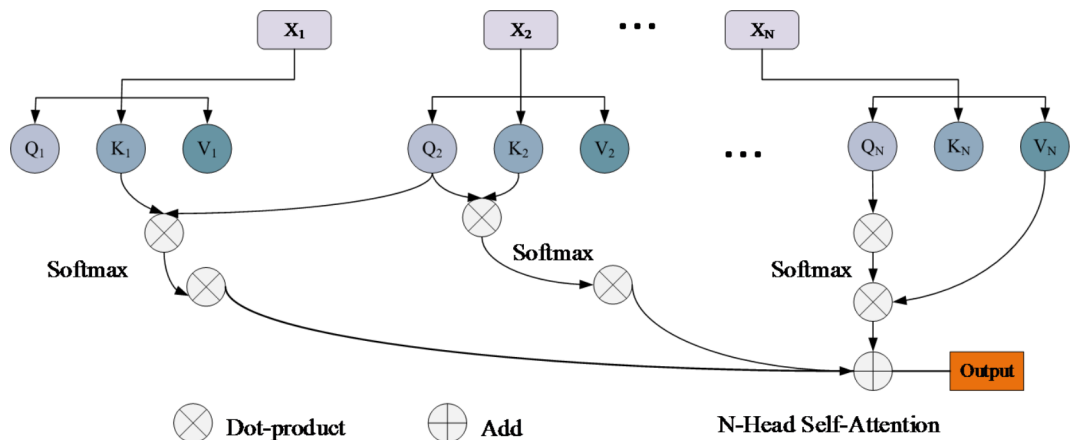


Fig. 5. The architecture of multi-head attention mechanism.



**Fig. 6.** Enumeration of datasets in different scenarios.

C2f_EffectiveSE	BiFPN_DoubleAttention	MHSA	Precision (%)	F1 (%)	mAP50 (%)
-	-	-	96.8	90.7	97.0
√	-	-	98.0	95.2	98.1
√	√	-	98.3	95.9	98.5
√	√	√	98.1	98.6	99.0

**Table 1.** Ablation experiment on each improved module.

$$Precision = \frac{TP}{TP + FP} \quad (6)$$

Recall is the ratio of the number of positive samples correctly predicted by the model to the number of actual positive samples, and is used to assess the comprehensiveness of the model's detection. The equation for recall is:

$$Recall = \frac{TP}{TP + FN} \quad (7)$$

The F1 score is the weighted average of precision and recall. The equation for the F1 score is defined as follows:

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall} = \frac{2TP}{2TP + FP + FN} \quad (8)$$

The mean average precision (mAP) is one of the key performance metrics used to evaluate target detection performance. It is the average of all average precision (AP) values in the dataset and describes the area under the precision-recall (P-R) curve. A higher mAP value indicates better detection performance of the model. The equation for mAP is:

$$mAP = \frac{1}{n} \sum_{i=1}^n \int_0^1 Precision(Recall) d(Recall) \quad (9)$$

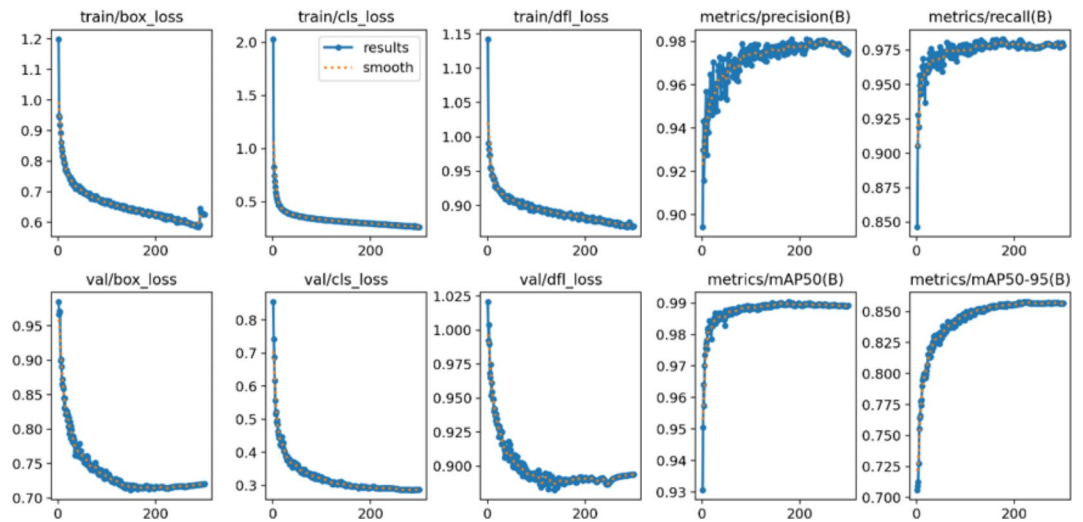
where C is the total number of classes and AP is the AP value of the *i*th class. In this study, two classes of target detection are used, so C=2. The average accuracy AP is calculated.

## Results and analysis

### Ablation experiment

To verify the effectiveness of the above improved algorithmic modules, ablation experiments were performed with different combinations of several modules, using the original YOLOv8n as the baseline model, with Precision, F1 score and mAP0.5 as the evaluation indexes. The experimental results are shown in Table 1. Ablation experiment on each improved module, and the training process is shown in Fig. 7.

The improved model's training process showed changes in accuracy and loss over iterations. As the number of iterations increased, the model updated its weights, resulting in increasing accuracy and decreasing loss. This indicates that within a certain range of iterations, more iterations led to the model learning more feature information and achieving higher accuracy. In the early stages of iteration, the loss decreased rapidly, and accuracy increased quickly. At around 100 iterations, mAP@0.5 stabilized at approximately 0.99, and the loss function reached a relatively stable state.



**Fig. 7.** Training result of the improved YOLOv8.

Method	Precision (%)	F1 (%)	Model size (MB)	mAP50 (%)	FPS/s
Mask-RCNN	88.7	83.9	300	86.8	28
DHT	89.0	81.1	130	91.7	35
UFLD	93.4	86.5	98	94.5	67
YOLOv5l	93.2	87.9	101	96.3	31
YOLOv8n	96.8	90.7	6.4	97.0	49
This paper	98.1	99.1	19.8	99.0	46

**Table 2.** Comparison of the performance of different algorithmic models.

#### Comparison with other algorithms

To verify the superiority of the algorithm proposed in this paper over currently popular conveyor belt runout detection algorithms, the algorithm of this paper is compared with several algorithms, namely Mask R-CNN, DHT, UFLD, YOLOv5, YOLOv8, and improved algorithm, under the same conditions in comparison experiments. The evaluation indices used are Precision, F1, Model size, mAP0.5 and FPS. The experimental results are shown in Table 2.

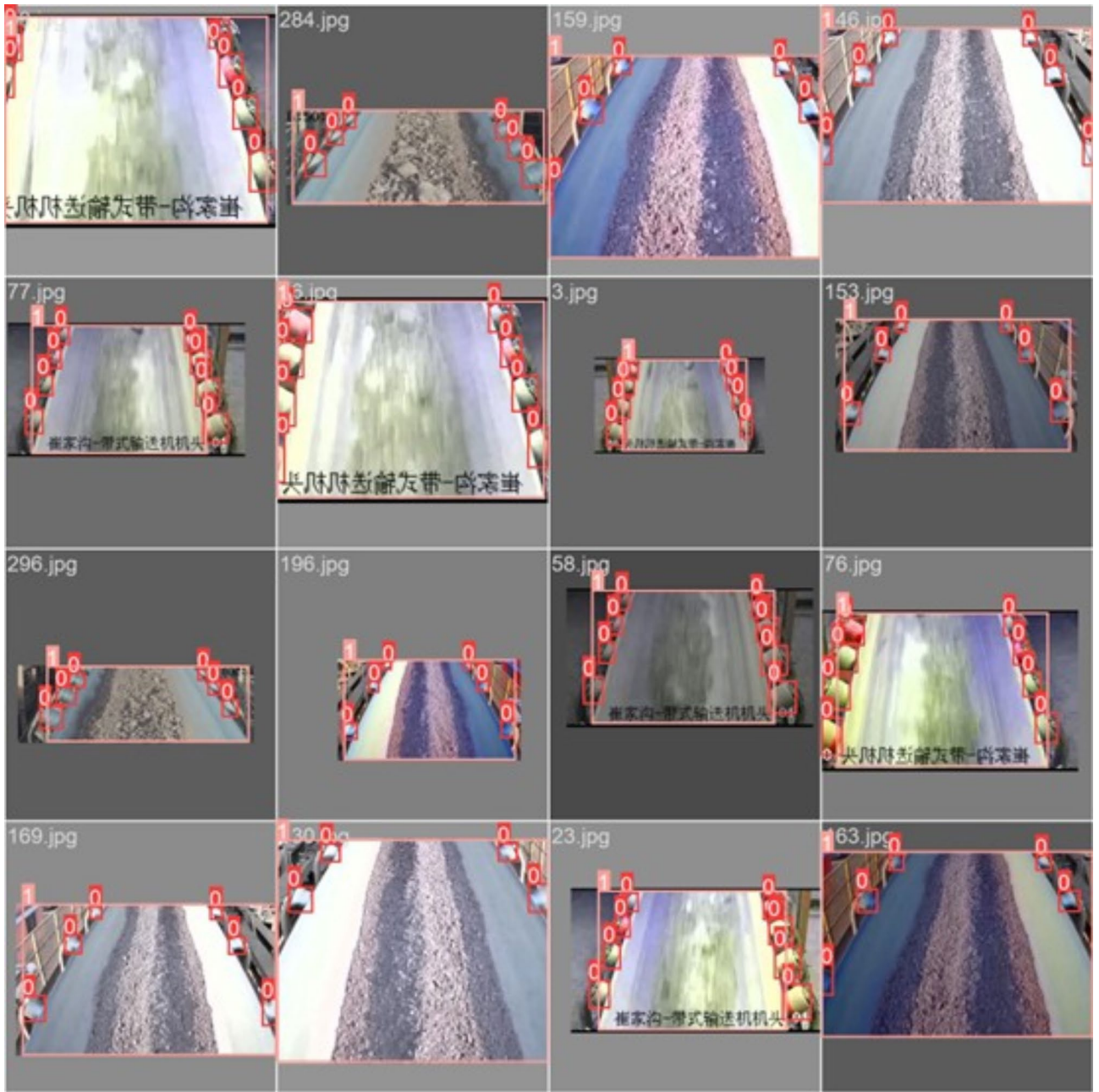
Compared to other conveyor belt detection algorithms, the improved YOLOv8 algorithm proposed in this paper demonstrates superior detection performance on our self-built conveyor belt oscillation dataset. This dataset encompasses diverse scenes, transmission states, and targets from various angles. Despite this diversity, the enhanced algorithm exhibits remarkable detection capabilities, achieving a mean Average Precision (mAP) of 99% at an IoU threshold of 0.5, with a detection speed of 46 frames per second (FPS) and a model size of 19.8 MB. Some test results are showcased in Fig. 8.

### Conveyor belt offset judgment method

Belt conveyors operate in three states: normal, left deviation, and right deviation. This paper presents a method for detecting conveyor belt misalignment by measuring the distance between the centerline of the trapezoidal area formed by the edge lines of the idlers and the centerline of the two outer edge lines of the roller supports and the centerline of the belt edge line, as shown in Fig. 9. The steps for data acquisition can locate the position of the belt and its edge lines in surveillance video images. Considering that the relative position between the camera and the belt conveyor remains unchanged after installation, and the outer edges of the idlers on both sides of the belt conveyor will form a straight line, it is easy to pre-determine the position of the idler edge lines in the image and store them in the program as reference lines. Under normal circumstances, the two previously mentioned centerlines coincide. Considering actual operation, the belt surface usually does not run strictly in the middle and may have slight left or right deviations. Therefore, a reasonable range of misalignment is allowed, with minor misalignments not considered faults. Threshold selection depends on working conditions and detection accuracy requirements.

#### Determination of the centerline

To determine the centerline of the edge line of the two roll holders, it is first necessary to clarify the definition of the roller bracket edge line equations. The left edge line equation of the roller holder can be expressed as:



**Fig. 8.** The roller and conveyor belt detection effect in different areas.

$$y_l = a_l x + b_l \tag{10}$$

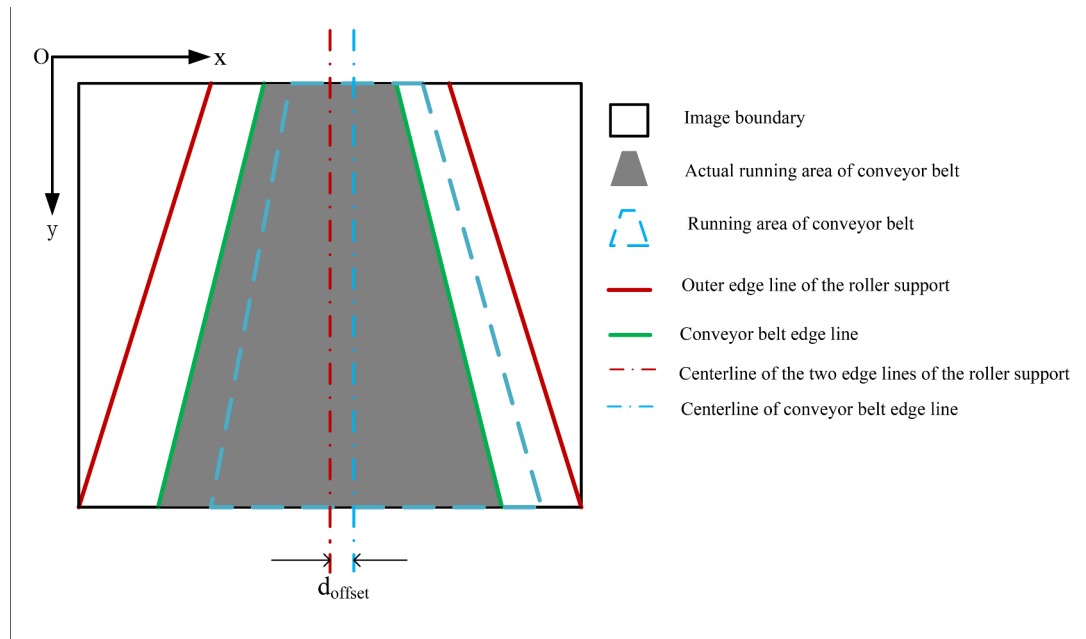
The equation for the right edge line of the roller support is expressed as:

$$y_r = a_r x + b_r \tag{11}$$

Here,  $x$  represents the horizontal coordinate in the image, corresponding to the pixel position;  $y_l$  and  $y_r$  denote the vertical coordinates of the left and right edge lines of the roller at position  $x$ , respectively. Parameters  $a_l$  and  $a_r$  are the slopes of the left and right roller edge lines, while  $b_l$  and  $b_r$  are the intercepts of the left and right roller edge lines.

The center line of the roller support region can be determined from the edge line equations. The equation of the center line is expressed as:

$$y_{roller\_center} = \frac{a_l + a_r}{2} x + \frac{b_l + b_r}{2} \tag{12}$$



**Fig. 9.** Offset judgment method.

$y_{roller\_center}$  represents the vertical coordinate of the center line of the roller support area at position  $x$ , indicating the central position of the roller support region. This centerline is derived by calculating the midpoint positions of the left and right roller holder outer edge lines and fitting these endpoint positions.

Following the same approach, the centerline equation for the conveyor belt is:

$$y_{belt\_center} = \frac{m_l + m_r}{2}x + \frac{c_l + c_r}{2} \tag{13}$$

In this equation,  $y_{belt\_center}$  represents the vertical coordinate of the centerline of the edge line of the conveyor belt at position  $x$ . This vertical coordinate signifies the central position of the conveyor belt, which is derived through the calculation and fitting of the midpoint positions of the edge lines.

**Calculate the deviation**

In conveyor belt deviation detection, the offset is defined as the perpendicular distance between the conveyor belt centerline and the centerline of the outer edges of the two idler brackets at the same  $x$ -coordinate. To calculate this offset, follow these steps: First, for a given  $x$ -coordinate, calculate the  $y$ -coordinate values of both the conveyor belt centerline and the idler centerline at that position. Then, by comparing the difference between these two  $y$ -coordinate values, the corresponding vertical distance is obtained, which is represented by the formula:

$$d_{offset} = |y_{belt\_center} - y_{roller\_center}| \tag{14}$$

Under normal circumstances, one can select multiple  $x$ -coordinate points to calculate the offset at different positions, and then take the average as the overall offset. The average offset can be expressed as:

$$\bar{d}_{offset} = \frac{1}{n} \sum_{i=1}^n |y_{belt\_center}(x_i) - y_{roller\_center}(x_i)| \tag{15}$$

where  $n$  represents the number of  $x$ -coordinate points selected,  $x_i$  represents the  $i$ -th  $x$  position.

Conveyor belt deviation judgment:

$$if \begin{cases} d > \tau, deviation \\ d \leq \tau, normal \end{cases} \tag{16}$$

$\tau$  serves as the threshold for determining the deviation of the conveyor belt.

## Conclusion

This paper proposes an improved conveyor belt deviation detection algorithm based on YOLOv8, addressing the challenges faced by traditional methods in objection detection applications. In the proposed approach, firstly, to enhance precise localization capabilities across multiple scales in complex environments, an efficient ESE module is integrated. Then a BiFPN\_DoubleAttention module is constructed in the neck network of YOLOv8, augmenting the feature extraction capabilities for multi-scale targets in complex environments. Thirdly, the introduction of a MHSA mechanism significantly improves detection accuracy for small target rollers. Finally, a deviation judgment method is then designed to enhance the generalization ability of our model.

Experiments demonstrate that our model has achieved higher detection accuracy than other existing models while meeting real-time requirements and reducing demand for computational and storage resources, making it easily deployable on resource-constrained devices. Future work will continue to expand the self-built dataset presented here, incorporating samples from different transportation environments to enhance industrial application potential.

## Data availability

The dataset used in this article is partly collected by ourselves, and partly selected and used from the dataset publicly available on GitHub. The URL of the dataset publicly available on GitHub is [https://github.com/zhangzhang1618/dataset\\_for\\_belt\\_conveyor/tree/master](https://github.com/zhangzhang1618/dataset_for_belt_conveyor/tree/master). If you would like the full dataset and code, please email the corresponding author.

Received: 11 June 2024; Accepted: 7 October 2024

Published online: 06 November 2024

## References

- Kiriia, R., Smirnov, A., Mostovyi, B., Mishchenko, T. & Novikov, L. Ways to reduce the risk of emergencies on belt conveyors operating in mining enterprises. in *IOP Conference Series: Earth and Environmental Science* Vol. 1348, No. 1, 012036 (IOP Publishing, 2024).
- Chu, Q. Y., Meng, G. Y. & Fan, X. Analysis of speed and belt deviation of the conveyor belt. *Adv. Mater. Res.* **339**, 444–447 (2011).
- Mei, X., Miao, C., Yang, Y. & Li, X. On-line intelligent evaluation of the fatigue state of a composite conveyor belt. *Therm. Sci.* **25**(3 Part B), 2191–2198 (2021).
- Nguyen, V., Jiao, R., Feng, D. & Hou, L. Study on running deviation mechanism and intelligent control of belt conveyor. *Vibroeng. Procedia* **36**, 115–120 (2021).
- Wang, X. J., Ding, Y. F., Liu, H. B., Chi, Y. L. & Xu, P. Study on running deviation and adjusting deviation of Belt Conveyor's structure based on Synchronous Technology. *Adv. Mater. Res.* **634**, 3794–3797 (2013).
- Yang, J. et al. Research on conveyor belt deviation detection method based on machine vision. In *Journal of Physics: Conference Series* Vol. 2786(1), 012013 (IOP Publishing, 2024).
- Yang, Y., Miao, C., Li, X. & Mei, X. On-line conveyor belts inspection based on machine vision. *Optik*, **125**(19), 5803–5807 (2014).
- Sun, L. & Sun, X. An intelligent detection method for conveyor belt deviation state based on machine vision. *Math. Model. Eng. Probl.* **11**(5) (2024).
- Fedorko, G. et al. Design of Evaluation Classification Algorithm for Identifying Conveyor Belt Mistracking in a Continuous Transport System's Digital Twin[J]. *Sensors* **24**(12), 3810 (2024).
- Wang, J., Liu, Q. & Dai, M. Belt vision localization algorithm based on machine vision and belt conveyor deviation detection. in *2019 34th Youth Academic Annual Conference of Chinese Association of Automation (YAC)* 269–273 (IEEE, 2019).
- Liu, Y., Wang, Y., Zeng, C., Zhang, W. & Li, J. Edge detection for conveyor belt based on the deep convolutional network. In *Proceedings of 2018 Chinese Intelligent Systems Conference* Vol. II, 275–283 (Springer, 2019).
- Xu, X., Zhao, H., Fu, X., Liu, M., Qiao, H., & Ma, Y. Real-time Belt deviation detection method based on depth edge feature and gradient constraint. *Sensors* **23**(19), 8208 (2023).
- Unlu, E., Zenou, E., Riviere, N. & Dupouy, P. E. Deep learning-based strategies for the detection and tracking of drones using several cameras. *IPSN Trans. Comput. Vis. Appl.* **11**, 1–13 (2019).
- Gao, M., Li, S., Chen, X. & Wang, Y. A novel combined method for conveyor belt deviation discrimination under complex operational scenarios. *Eng. Appl. Artif. Intell.* **137**, 109145 (2024).
- Zhao, X., Zeng, M., Dong, Y., Rao, G., Huang, X. & Mo, X. FastBeltNet: a dual-branch light-weight network for real-time conveyor belt edge detection. *J. Real-Time Image Process.* **21**(4), 123 (2024).
- Włodarczyk-Sielicka, M. & Polap, D. Automatic classification using machine learning for non-conventional vessels on inland waters. *Sensors* **19**(14), 3051 (2019).
- Long, J., Shelhamer, E. & Darrell, T. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* 3431–3440 (2015).
- Chen, L. C., Papandreou, G., Kokkinos, I., Murphy, K. & Yuille, A. L. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **40**(4), 834–848 (2017).
- Xie, S., & Tu, Z. Holistically-nested edge detection. In *Proceedings of the IEEE international conference on computer vision* 1395–1403 (2015).
- Zeng, C., Zheng, J. & Li, J. Real-time conveyor belt deviation detection algorithm based on multi-scale feature fusion network. *Algorithms* **12**(10), 205 (2019).
- Zhang, M. et al. A deep learning-based method for deviation status detection in intelligent conveyor belt system. *J. Clean. Prod.* **363**, 132575 (2022).
- Wang, R. et al. LightR-YOLOv5: A compact rotating detector for SARS-CoV-2 antigen-detection rapid diagnostic test results. *Displays* **78**, 102403 (2023).
- Hu, H. et al. Road surface crack detection method based on improved YOLOv5 and vehicle-mounted images. *Measurement* **229**, 114443 (2024).
- Mathew, M. P. & Mahesh, T. Y. Leaf-based disease detection in bell pepper plant using YOLO v5. *Signal Image Video Process.* 1–7 (2022).
- Sun, X., Wang, Y. & Meng, W. Evaluation system of curved conveyor belt deviation state based on the ARIMA-LSTM combined prediction model. *Machines* **10**(11), 1042 (2022).
- Wu, L., Wang, Y., Zhang, W., Huang, S. & Li, J. X. *Research on Belt Deviation Diagnosis of Belt Conveyors Based on Deep Learning* (2024).

27. Sun, S. et al. Multi-YOLOv8: An infrared moving small object detection model based on YOLOv8 for air vehicle. *Neurocomputing* **588**, 127685. (2024).
28. Wan, D. et al. YOLO-MIF: Improved YOLOv8 with Multi-Information fusion for object detection in Gray-Scale images. *Adv. Eng. Inform.* **62**, 102709 (2024).
29. Xiao, B., Nguyen, M. & Yan, W. Q. Fruit ripeness identification using YOLOv8 model. *Multimedia Tools Appl.* **83**(9), 28039–28056 (2024).
30. Liu, S., Qi, L., Qin, H., Shi, J. & Jia, J. Path aggregation network for instance segmentation. in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* 8759–8768 (2018).
31. Chen, T. et al. Detection network for multi-size and multi-target tea bud leaves in the field of view via improved YOLOv7. *Comput. Electron. Agric.* **218**, 108700 (2024).
32. Zhou, W., Cai, C., Zheng, L., Li, C. & Zeng, D. ASSD-YOLO: a small object detection method based on improved YOLOv7 for airport surface surveillance. *Multimed. Tools Appl.* **83**(18), 55527–55548 (2024).
33. Lu, C., Xia, Z., Przystupa, K., Kochan, O. & Su, J. DCELANM-Net: Medical image segmentation based on dual channel efficient layer aggregation network with learner. *Int. J. Imaging Syst. Technol.* **34**(1), e22960 (2024).
34. Li, X. et al. Generalized focal loss: Learning qualified and distributed bounding boxes for dense object detection. *Adv Neural. Inf. Process. Syst.* **33**, 21002–21012 (2020).
35. Zheng, Z. et al. Enhancing geometric factors in model learning and inference for object detection and instance segmentation. *IEEE Trans. Cybernet.* **52**(8), 8574–8586 (2021).
36. Chen, G. et al. YOLOv8-CML: A lightweight target detection method for Color-changing melon ripening in intelligent agriculture. *Sci. Rep.* **14**(1), 14400 (2024).
37. Sun, Y. et al. SES-YOLOv8n: automatic driving object detection algorithm based on improved YOLOv8. *Signal. Image Video Process.* **18**(5), 3983–3992 (2024).
38. Du, X. et al. Research on Fault Detection of BeltConveyor Drum Based on Improved YOLOv8 Network Mode (2024).
39. Chen, K., Du, B., Wang, Y., Wang, G., & He, J. The real-time detection method for coal gangue based on YOLOv8s-GSC. *J. Real-Time Image Proc.* **21**(2), 37 (2024).
40. Bajpai, A., Tiwari, N., Yadav, A., Chaurasia, D., & Kumar, M. Enhancing underwater object detection: Leveraging YOLOv8m for improved subaquatic monitoring. *SN Comput. Sci.* **5**(6), 1–10 (2024).
41. Xing, Z., Zhu, Y., Liu, R., Wang, W., & Zhang, Z. DCM-YOLOv8: An improved YOLOv8-based small target detection model for UAV images. In *International Conference on Intelligent Computing*, 367–379 (Springer Nature Singapore, Singapore, 2024).
42. Zhao, X. Y. et al. A quality grade classification method for fresh tea leaves based on an improved YOLOv8x-SPPCSPC-CBAM model. *Sci. Rep.* **14**(1), 4166 (2024).
43. Lee, Y. & Park, J. Centermask: Real-time anchor-free instance segmentation. in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* 13906–13915 (2020).
44. Hu, J., Shen, L. & Sun, G. Squeeze-and-excitation networks. in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* 7132–7141 (2018).
45. Tan, M., Pang, R. & Le, Q. V. Efficientdet: Scalable and efficient object detection. in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* 10781–10790 (2020).
46. Chen, Y., Kalantidis, Y., Li, J., Yan, S. & Feng, J. *A<sup>2</sup>-nets: Double Attention Networks* 31 (Advances in Neural Information Processing Systems, 2018).
47. Srinivas, A. et al. Bottleneck transformers for visual recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* 16519–16529 (2021).

## Author contributions

All the authors contributed extensively to the manuscript. Yunfeng Ni. conceived and conducted the experiments, Haixin Cheng wrote the main manuscript, and other authors made revisions and suggestions to the paper. All authors have read and agreed to the publication of the manuscript.

## Funding

This work was supported in part by the National Natural Science Foundation of China under Grant 61701393.

## Declarations

## Competing interests

The authors declare no competing interests.

## Additional information

**Correspondence** and requests for materials should be addressed to H.C.

**Reprints and permissions information** is available at [www.nature.com/reprints](http://www.nature.com/reprints).

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

**Open Access** This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by-nc-nd/4.0/>.

© The Author(s) 2024